CSCI 3353 Object Oriented Design
Homework Assignment 3
Due Monday, February 12

This assignment asks you to write two programs and a design document.

1. The first program is called *HW3DiceGame*, which is a new version of the HW 2 dice game having the following five features:

A. The *Player* interface should be an abstract class containing the common code from the three player classes.

B. The program should hold a map of all players, keyed by player name. This map is stored in a file. It gets loaded into memory when the program is executed, and saved back to the file when the program finishes. This map is initially empty. When a player participates in a game for the first time, its *Player* object gets added to the map (and will be saved to the file when the game is over).

The name of the file can be anything. Please choose a name that will not be chosen by anyone else in the class. Examples of doing a load/save to a file are in the Chapter 3 notes. Don't forget about the *Serializable* interface!

C. When a game is being set up, the user should specify the name of each player. If the map contains a player having that name, then its *Player* object is retrieved from the map. Otherwise, the program asks the user for the type of the player (human, crafty, or timid) and creates the appropriate *Player* object to use. It then adds this object to the map.

D. Each *Player* object should keep track of that player's wins and losses. Its *toString* method should return a string containing the player name, its type ("Human", "Crafty Bot", or "Timid Bot"), its wins, losses, and its winning percentage. Winning percentage should be an integer. For example, the winning percentage of a player having 2 wins and 1 loss is 66.

The *toString* methods for the three kinds of player will be very similar. Use the template pattern to avoid duplicating code. That is, the code for *toString* will be defined in *Player*, and will call an abstract helper method implemented by the three subclasses.

E. Player objects should be comparable by their winning percentage, in descending order. That is, p1 < p2 if p1 has a higher winning percentage than p2. In the case of a tie, compare by the player names, ascending.

2. The second program is called *HW3PlayerStats*. This program should load the player map from its file, and print the information about each player in sorted order. (That is, the player having the highest winning percentage is printed first.)

For printing out objects in sorted order, consider using a *TreeSet* or the method *Collections.sort*.

3. Write a document, named *HW3Design.pdf*, that explains the changes you made. In particular, I want you to describe any new classes you created (if any), and for all existing classes, what new state variables you added and what new methods you wrote. Explain the purpose of each new variable or method if it is not obvious.

If you are uncomfortable using your HW2 code for this assignment, email me and I will send you my solution to use. In either case, be sure to copy all the java files to the package *hw3*.

Please submit your design document and all java files needed to run both programs. Please submit individually; do not zip them.